3T

УДК 004.921, 004.928

Б.А. Троценко
Київський національний університет імені Тараса Шевченка, м. Київ, Україна
modosansreves@gmail.com

Дослідження інформаційних процесів для ефективного відтворення дактильної жестової мови

У статті розглядається підхід до показу дактилем української дактильної жестової мови за допомогою просторової моделі. Розглядаються математична модель скелета руки людини, алгоритми анімації руки за допомогою скелетної моделі, алгоритми переходу між дактилемами у процесі показу слова, алгоритми підготовки поверхні руки (скінінгу), алгоритм розпаралелення обчислень у підзадачі підготовки кадрів для анімації показу.

Вступ і постановка задачі

Зі створенням незалежної української держави та визнанням на державному рівні жестової мови як засобу міжособистісного спілкування, так і засобу навчання нечуючих осіб виникла нагальна потреба у вивченні національної жестової мови та створенні сучасних словників і засобів навчання цій мові.

Як і властиво мові, жестова мова має діалекти. На відміну від усного мовлення, яке використовується на телеканалах, радіо, а через буквений запис – у газетах та книгах, жестова мова не має писемного еквівалента, рідко використовується на телебаченні. Як результат, жестова мова має діалектичні відмінності навіть у межах області. Наявність певного «еталона» жестових одиниць (дактилем) у запису дозволить полегшити проблему діалектів.

Наразі жестову мову можна вивчати за допомогою вчителя, книги або відеозапису. Але кожен вчитель має «свою» дактильну жестову мову, його аудиторія навчання обмежена, вчитель як «медіаносій» не копіюється, не можна повторити/побачити жест поза навчанням. Стосовно навчання за допомогою книги [1]: двовимірне зображення не передає динаміки та деталей жестових одиниць, книга не є інтерактивним медіаносієм. Навіть більш сучасний спосіб подання жесту за допомогою відеозображення [2] – це: великий об’єм даних, жест не можна розглянути з різних сторін, із окремих записаних жестів не можна скомпонувати речення, яке б виглядало природно, відео не є інтерактивним медіаносієм.

Сучасний розвиток комп’ютерної техніки спонукає запропонувати більш ефективний підхід (медіазасіб зображення) до навчання жестовій мові, зокрема дактильній жестокій мові, що розглядається у даній роботі. Використання комп’ютерної технології сприятиме спорідненню діалектів, легкість копіювання програм та можливість використати мережу Інтернет [3] для віддаленого показу дактилем на комп’ютерах різної потужності сприятимуть поширенню знань і навичок володіння дактильною жестовою мовою.

Постановка задачі. Запропонувати алгоритм та реалізацію оптимального за часом розрахунку стану просторової моделі за умови її високої розмірності, наявності скелета, близького до скелета руки людини, та звичайних мультимедійних вимог до апаратної частини комп’ютера, розглянути алгоритми розрахунку поверхні моделі з метою виявлення найефективнішого, запропонувати метод показу дактильованого слова на основі окремих дактилем, для обчислень використати усі ядра процесора. Під оптимальним за часом розрахунком мається на увазі час, що відповідає частоті зміни кадрів 30 за секунду і більше.

Математична модель

Для вирішення задачі використаємо скелетну модель руки. Скелет руки математично зобразимо у вигляді дерева (V,E), де V – множина вершин, E – множина з'єднань. Кожна вершина відповідає деякому зчленуванню кісток (ланок скелета). Кожен елемент $(e_i,e_j)\in E : e_i,e_j \in V$ відповідає за зв'язок між ланками, при цьому ланку e_i називатимемо *старшою* по відношенню до ланки e_j, а ланку e_j – *дочірньою* по відношенню до e_i. Кожна ланка може мати не більше однієї старшої і довільну кількість дочірніх ланок. Ланку, що не має відповідної їй старшої, називатимемо *основною*.

За $P(e)$ будемо позначати старшу по відношенню до e ланку, якщо така існує, і 0 – в іншому випадку.

Для кожної ланки $e\in V$ поставимо у відповідність впорядковану двійку $h(e)=(x,\theta)$, де $x\in R^3$ – положення зчленувань ланки зі старшою у системі координат, пов'язаній з $P(e)$. Параметр θ задає порядок застосування кутів повороту для перетворення системи координат, що пов'язана з ланкою, $h(V)$ задає початкове положення руки для математичної моделі (V,E).

Для кожної ланки $e\in V$ поставимо у відповідність вектор $r(e)\in R^3$, що складається з кутів повороту навколо осей *OX*, *OY*, *OZ*. Таким чином, сукупність $r(V)$ задає положення руки, дану сукупність також називатимемо *параметрами моделі*. За наявності відношення порядку між елементами множини вершин (V), наприклад, за індексом, $r(V)$ можна єдиним чином представити у вигляді матриці розмірності $3\times|V|$, де i-й стовпець є вектором $r(e_i), e_i \le e_{i+1}$. Дану матрицю називатимемо *матрицею параметрів моделі*.

Основною операцією у даній математичній моделі є отримання, на основі параметрів моделі, положень точок з'єднань ланок і матриць переходу систем координат відносно системи координат, що відповідає основній ланці.

Матриці переходу систем координат визначаються формулами:

$$M(e)=\begin{cases} M(P(e))\cdot f(\theta, M_X^{r(e)}, M_Y^{r(e)}, M_Z^{r(e)}), P(e)\ne 0 \\ I,\ P(e)=0 \end{cases},$$

де $M_X^{r(e)}$ – матриця повороту системи координат навколо осі *OX* на кут, що записаний у першому рядку вектора $r(e)$, $M_Y^{r(e)}$ – матриця повороту системи координат навколо осі *OY* на кут, що записаний у другому рядку вектора $r(e)$, $M_Z^{r(e)}$ – матриця повороту системи координат навколо осі *OZ* на кут, що записаний у третьому рядку вектора $r(e)$, I – одинична матриця розмірності 3х3, $f(\theta, M_1, M_2, M_3)$ – добуток матриць M_1, M_2, M_3 у порядку, визначеному θ.

Положення ланки у глобальній системі координат визначається за формулою:

$$p'(e)=x(e)\cdot M(e),$$

де $p'(e)$ – положення ланки у глобальній системі координат для стану системи M.

На основі даної математичної моделі побудуємо інформаційну модель руки, що складатиметься з математичної моделі, матриці точок у просторі для моделі у початковому положенні, текстури поверхні руки, матриці текстурних координат, матриці нормалей.

Отримання візуального зображення руки у заданому положенні у певний момент часу відбувається за допомогою процедури скінінгу (розрахунку точок поверхні шкіри) на основі положень ланок у глобальній системі координат. Кожна точка має відповідно одну або більше ланок, що впливають на розташування точки з певним коефіцієнтом.

Сума коефіцієнтів залежності для кожної точки складає 1. Цей підхід дозволяє уникнути візуально неприродних частин у місцях, близьких до зчленувань ланок, наприклад, у фалангах пальців. Без використання подібного згладжування за допомогою коефіцієнтів у практичному застосуванні можливо спостерігати «розриви» видимої моделі, вид зсередини на частини ланок, що оперуються скелетною моделлю.

Для отримання матриць параметрів моделі було використано технологію Motion Capture (захоплення руху). Дана технологія передбачає використання оцифрованих даних про рух, що записано з людини-носія жестової мови. Для цього існує спеціальна рукавиця, що має яскраві зони в областях, близьких до з'єднань кісток. На кілька відеокамер записується «промовляння» усієї абетки дактильною жестовою мовою. За допомогою спеціального програмного забезпечення, Motion Builder, отримується динаміка зміни положень у просторі виділених зон, а на основі цих даних – інформація про зміну кутів.

Таким чином, за рахунок використання даних такої природи немає необхідності у введенні обмежень на множину допустимих кутів у математичній моделі. Дані, отримані за допомогою технології Motion Capture, більш реалістично відтворюють оригінальну динаміку «промовляння» дактилеми, ніж дані, отримані за допомогою дизайнерського програмного забезпечення.

Алгоритми розрахунку точок поверхні руки (скінінг)

Поверхня моделі руки є множина точок, положення яких залежне від положень ланок. Кожній точці зіставлено множину $\{(e_i, w_i)\}$, де e_i – ланка, що впливає на точку, $w_i \in [0,1]$ – коефіцієнт залежності, $\Sigma_i w_i = 1$. Відповідно положення точки у просторі визначається формулою $\Sigma_i p'(e_i) \cdot w_i$. Алгоритм розрахунку даної формули для усіх точок інформаційної моделі та підготовка відповідних структур даних називаються *скінінгом* [4].

Для ефективної візуалізації моделі на практиці було запропоновано декілька алгоритмів скінінгу з метою подальшої апробації. Усі алгоритми використовують масив вхідних даних точок у початковому положенні, записують результат у масив точок для результуючого положення. Типово, що масив результату ініціалізується точками з координатами (0, 0, 0), у процесі роботи алгоритму до відповідних елементів додаються складові частини; таким чином, лише наприкінці роботи алгоритму результуючий масив містить правильні положення точок.

Послідовний алгоритм скінінгу використовує подання залежностей у вигляді списків (ланка; масив коефіцієнтів залежностей, кожен елемент якого відповідає елементу з масиву координат точок у початковому положенні). Він описується наступним псевдокодом:

```
для кожної ланки i:
    для кожної вершини j:
        отримати координату вершини j у початковому положенні,
        домножити на матрицю переходу у систему координат,
         що пов'язана ланкою i,
        домножити на коефіцієнт залежності,
        додати координату до елементу масиву результатів.
```

Алгоритм розраховано на те, що локальні частини масивів координат початкового положення, коефіцієнтів залежностей, результуючих координат на практиці перебуватимуть у кеш-пам'яті процесора, що дозволить швидку обробку даних з відносно малою кількістю операцій обміну даними з оперативною пам'яттю.

Послідовний алгоритм скінінгу *з міні-макс оптимізацією* є покращеним варіантом алгоритму послідовного алгоритму скінінгу, де на етапі передобробки для кожної ланки обчислюються мінімальний та максимальний індекси вершин, на які дана ланка впливає. Це орієнтовано на те, що залежні точки займають у масиві координат приблизно сусідні індекси, що дозволить зменшити кількість операцій над вершинами, коефіцієнт залежностей яких є 0.

Сегментаційний алгоритм скінінгу є покращеним варіантом алгоритму послідовного алгоритму, використовує подання у вигляді списків (ланка; список пар (початковий індекс, масив коефіцієнтів залежностей)). Подання вибирається таким чином, що довжина масиву коефіцієнтів залежностей не перевищує задану, наприклад 256. Кожен елемент з індексом *i* з масиву коефіцієнтів залежностей відповідає елементу *i + початковий індекс*. Розрахований на оптимальне використання кеш-пам'яті процесора у практичному використанні. Він описується наступним псевдокодом:

```
для кожної ланки i:
    для кожної пари зі списку
    (початковий індекс, масив коефіцієнтів залежностей),
    що відповідає ланці i:
        для кожного коефіцієнта залежностей з масиву
        з індексом k:
            отримати координату вершини у початковому положенні
             з індексом (k + початковий індекс),
            домножити на матрицю переходу у систему координат,
             що пов'язана ланкою i,
            домножити на коефіцієнт залежності,
            додати координату до елементу масиву результатів
             з індексом (k + початковий індекс)
```

Прямий алгоритм використовує подання залежностей у вигляді списків пар (ланка; масив пар (індекс вершини; коефіцієнт залежності)). Він використовує рекурсивну процедуру обходу вершини, що описується наступним псевдокодом. Сам алгоритм полягає у виконанні процедури обходу старшої вершини.

```
процедура Обхід вершини (ланка i)
    для кожної пари з масиву, що відповідає ланці i:
        отримати координату вершини
         у початковому положенні за індексом,
        домножити на матрицю переходу у систему координат,
         що пов'язана ланкою i,
        домножити на коефіцієнт залежності,
        додати координату до елементу масиву результатів
для кожної ланки j, що є дочірньою до ланки i:
        виконати обхід вершини j
```

Прямий алгоритм на практиці виконує мінімальну кількість операцій серед наведених. У випадку, коли сума об'ємів пам'яті, що займають точки моделі, масив координат початкових положень, масив координат результуючих положень, додаткові структури даних менше, ніж об'єм кеш-пам'яті процесора, даний алгоритм показує найвищу ефективність. В іншому випадку рекомендується використовувати сегментаційний алгоритм.

Дані алгоритми було імплементовано, проведено порівняльні тести на ефективність. Під ефективністю алгоритму вважається швидкість розрахунку окремого кадру, час на підготовку структур даних ігнорується. Тестування проводилось на скелетній моделі, що має 84 460 точок на комп'ютері з процесором Intel Core2 Duo E8200 (2,66 ГГц), 2 Гб оперативної пам'яті. Розрахунок проводився в один потік. Результати тестування наведено на рис. 1.

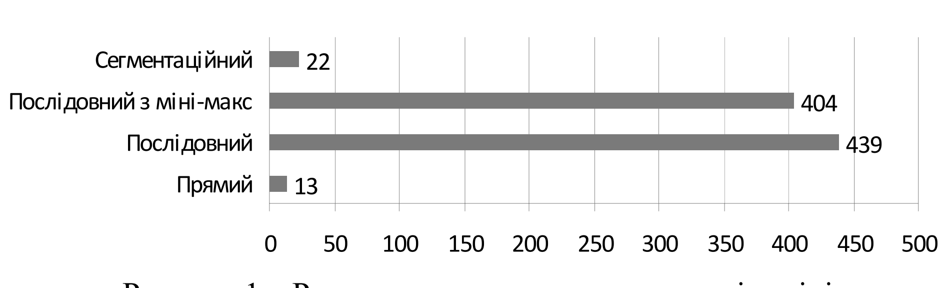


Рисунок 1 – Результати тестування алгоритмів скінінгу

Алгоритм дактилювання слова як конкатенація окремих дактилем

Важливою перевагою даної системи є можливість дактилювати довільні слова, що складаються з окремих дактилем («літер»). За наявності інформації про окремі дактилеми важливо мати спосіб їх візуальної конкатенації. З використанням моделі руки як основи подання графічної інформації це є алгоритмічно можливим, на відміну від даних у відеоформаті.

Було створено базу даних, що містить послідовності станів моделі для кожної літери. Кожна така послідовність починається і закінчується у деякому «нейтральному» стані. Для правильної конкатенації потрібним є підхід для визначення граничних станів кожної дактилеми, де закінчується інформація, що важлива для передачі змісту дактилеми, та алгоритм створення проміжних станів для переходу між дактилемами.

Зважаючи на природу отриманих даних, граничні стани були встановлені експериментально. Генерування проміжних станів може бути або алгоритмічним (на основі станів, що конкатенуються), або наперед визначеним. Алгоритмічний підхід є більш зручним, оскільки зменшує кількість необхідної інформації, що потрібно підготувати дизайнеру. Для більшості пар дактилем можна алгоритмічно згенерувати проміжні слайди, наприклад, лінійною апроксимацією на основі граничних станів у суміжних літерах.

Є послідовності, наприклад, «Н» – «Г» [1], де потрібно спершу зігнути пальці, а потім повертати долоню, для яких не підходить алгоритмічний підхід. Для цього випадку передбачається використання окремої послідовності станів переходу.

Розпаралелення обчислень у задачі підготовки кадру

Більшість сучасних процесорів, що випускаються, оснащено двома або більше ядрами. Подальший розвиток орієнтований саме на збільшення кількості ядер, ніж на нарощування потужності окремого ядра [5].

Для підзадачі підготовки кадру в задачі показу тривимірної анімації дактильної абетки жестової мови було розглянуто алгоритм розпаралелювання обчислень, оскільки за своєю суттю підготовка одного кадру є функцією від моделі руки і параметрів моделі, тобто не залежить від результату обчислень інших кадрів і від моменту обчислення. Такий підхід має місце, коли обчислення нового кадру є ресурсномісткою задачею, наприклад, проведення скінінгу на процесорі. Для цього був розроблений алгоритм, що складається з $2 + n$ потоків, де n – кількість ядер процесора: основний потік, потік анімації, n потоків обчислення кадрів. Використовується факт, що обчислення різних кадрів є рівносильною задачею в термінах процесорного часу незалежно від параметрів моделі. Тому обчислення k кадрів можна розбити на n потоків максимально рівномірно таким способом: кожен i-й потік обробляє кадри з тим номером, залишок від ділення якого на n дорівнює i.

Основний потік (потік інтерфейсу користувача) ініціює створення n потоків обчислень і потоку анімації. Потік анімації показує із заданим проміжком підготовлені кадри, звільняє пам'ять після їх показу. Оскільки на момент початку анімації повинна бути впевненість, що кожен кадр буде готовим перед його показом, то потік інформації запускається призупиненим, а його запуск делегується на один з потоків обчислення кадрів, зокрема на потік з індексом 0. Надалі цей потік називатимемо *ведучим*.

Кожен потік обчислення кадрів на початку отримує залишок від ділення і починає цикл обчислень. Також проводиться оновлення лічильника обчислених кадрів.

Ведучий потік також фіксує момент часу початку обчислень. Після того, як було обчислено 5 кадрів, він апроксимує момент часу закінчення обчислень усіх кадрів. Оскільки тривалість анімації відома спочатку, то можливо визначити, чи буде підготовка

кадрів виконана вчасно для безперебійного показу анімації. Коли апроксимований час до кінця закінчення обчислень стає меншим за тривалість анімації, ведучий потік запускає потік анімації і припиняє прогнозування закінчення.

Задача розподілення потоків на кожне ядро виконується операційною системою. Але часто бувають ситуації, коли ядра зайняті сторонніми процесами. Це призвело б до того, що потоки передобробки отримали різні кванти процесорного часу, а як наслідок, кадри обчислювались нерівномірно.

Наприклад, можливою була б така ситуація (у прикладі використовується двоядерний процесор):

Таблиця 1

№ кадру	0	1	2	3	4	5	6	7	...
Стан	+	+	+	–	+	–	+	–	...

Де стан «+» означає, що кадр підготовлено, «–» – не підготовлено.

Використано підхід вирішення даної проблеми за допомогою семафорів [6], [7]: кожен потік проасоційовано з одним семафором, який ініціалізовано одиницею. На початку обчислення кадру потік зменшує свій семафор, після завершення кадру він збільшує семафор наступного потоку на одиницю (останній потік збільшує семафор нульового потоку). Таким чином, обчислення кадрів одного потоку не випереджає решту більше, ніж на одиницю.

За своїм призначенням, значення семафора не може змінюватись менше за нуль. Це є об’єкт ядра операційної системи, який відповідає за сумісний доступ кількох потоків до спільного ресурсу. У цього об’єкта дві операції – збільшити (на одиницю), зменшити. Якщо значення семафора – нуль, а потік виконує операцію «зменшити», виконання його призупиняється операційною системою. А отже, стає доступним додатковий процесорний час. Операційна система веде облік усіх потоків, що були призупинені внаслідок спроби зменшити семафор. Коли деякий інший робить операцію «збільшити», то інший потік зі списку призупинених продовжує виконання.

Практично такий підхід означає наступне: кожен потік анімації на початку роботи алгоритму «може» обчислити 1 кадр, за що відповідає значення його семафора. Після обчислення свого кадру він дозволить обчислення іншому потокові і буде призупиненим, аж поки його семафор не збільшить потік. Таким чином, ядра процесора максимально навантажені, а кадри обчислюються поступово.

Використання технології передачі даних через мережу Інтернет

Для поширення програмного додатка, який дозволяє дактилювати окремі літери та слова, використано узагальнений засіб оптимальної передачі динамічно створеного зображення в мережі Інтернет з можливістю керування ним [4]. Оскільки обчислювальні можливості комп’ютерів широкого кола користувачів значно різняться, так само як і значно різняться можливості їх каналів зв’язку, даний аспект було враховано при продумуванні способу масового обслуговування, для передачі даних через мережу Internet. Наявні технології або не мають інтерактивності, або для її досягнення потрібно використовувати спеціальні плагіни, які необхідно інсталювати на комп’ютер клієнта.

Технологія передачі даних містить трирівневу інформаційну модель комунікації між серверною та клієнтськими частинами: на першому (найвищому) логічному рівні відеоінформація передається зі серверної сторони на сторону клієнта. На наступному, другому, логічному рівні сервер закодовує відеоінформацію фрейм за фреймом, а також може передбачати певну ініціалізацію клієнта, клієнт, проводить ініціалізацію, а також відтворює картинку на основі команд. На цьому логічному рівні програміст (інженер,

розробник) може враховувати специфіку задачі і структуру медіаданих, тобто надати власну пару компресора-декомпресора даних. На цьому рівні окремий потік відповідає за обслуговування іменних каналів та передачу команд компресору. Рекомендований рівень компресії виходить з розрахунку, що 10 відсотків часу канал повинен або простоювати, або резервуватися на непередбачену ситуацію, наприклад, передачу даних іншим програмним додаткам на стороні клієнта. На наступному, третьому, логічному рівні команди серіалізуються на стороні сервера і десеріалізуються на стороні клієнта. Це є останній логічний рівень даної технології. Інформація на даному рівні передається використанням HTTP (Silverlight/Asp.net).

Висновки

У даній статті вперше для відтворення дактильної української жестової мови використано анімацію тривимірної моделі руки, розглянуто алгоритми розрахунку поверхні руки за допомогою тривимірної моделі, що базується на скелетній моделі, виконано реалізацію цих алгоритмів та практичну апробацію, розглянуто алгоритм анімації дактилювання слова на основі даних про окремі дактилеми, удосконалено алгоритми підготовки кадру для відтворення процесу скелетної анімації дактильної жестової мови за рахунок підтримки багатоядерних процесорів. Дані алгоритми та методи є сумісними з технологією відображення інформації інтернет-засобами для відтворення процесу анімації дактильної жестової мови, що дозволить створити відповідний програмний додаток.

Подальші дослідження будуть направлені на підвищення якості відображення руки, покращення алгоритмів показу з метою підтримки використання програми на комп'ютерах з меншою потужністю.

Література

1. Кульбіда С.В. Українська дактилологія / Кульбіда С.В. – К. : Педагогічна думка, 2007. – 255 с.
2. Воскресенский А.Л. От звучащей речи к жестовой / А.Л. Воскресенский, Г.К. Халагин // Речевые технологии. – 2009. – № 1. – С. 99-106.
3. Бармак О.В. Технологія оптимізованої передачі жестової мови в мережі Інтернет / О.В. Бармак, Ю.В. Крак, Б.А. Троценко // Проблеми програмування. – 2009. – № 3. – С. 73-80.
4. Domine S. Mesh Skinning / S. Domine [Електронний ресурс]. – Режим доступу : http://developer.nvidia.com/attach/6662. – Назва з екрану
5. Merritt R. CPU designers debate multi-core future / R. Merritt [Електронний ресурс]. – Режим доступу : http://www.eetimes.com/showArticle.jhtml?articleID=206105179. – Назва з екрана.
6. Agarwal A. Partitioning Strategies for Concurrent Programming / Anant Agarwal, Srinivas Devadas, Henry Hoffmann // Technical Report MIT-CSAIL-TR-2009-026. – 2009. [Електронний ресурс]. – Режим доступу: http://dspace.mit.edu/bitstream/handle/1721.1/45567/MIT-CSAIL-TR-2009-026.pdf. – Назва з екрана.
7. Massingill B. Patterns for parallel programming / Massingill B., Mattson T., Sanders B. – Reading, Massachusetts : Addison-Wesley Professional, 2004. – 384 p.

Б.А. Троценко
Исследование информационных процессов для эффективного отображения дактильной жестовой речи
В статье рассматривается подход представления дактилей украинского дактильно-жестового языка с помощью пространственной модели. Рассматривается математическая модель скелета руки человека, алгоритмы анимации руки с помощью скелетной модели, алгоритмы перехода между дактилемами в процессе представления слова, алгоритмы подготовки поверхности руки (скининг), алгоритм параллельных вычислений в подзадачах подготовки кадров для анимации показа.

B.A. Trotsenko
An Investigation Information Processes for Effective Visualization of Fingerspelling
One of the approaches to visualize a hand during the fingerspelling process is proposed in the paper. Appropriate mathematical skeleton-based model of a hand is suggested. Algorithms for hand animation using the model are explored. Skinning algorithms are researched and compared. An algorithm for proper transition of hand states during the fingerspelling of a word is suggested. The parallel algorithm of computing frames for the animation is considered.

Стаття надійшла до редакції 13.07.2010.